Une méthode efficace pour éviter la propagation des fake news

Silvia Bonomi, Giovanni Farina, Sébastien Tixeuil

Une méthode efficace pour éviter la propagation des fake news

Silvia Bonomi[1] et Giovanni Farina [12 †] et Sébastien Tixeuil[2]

[1] *Sapienza Università di Roma, Rome, Italy*
[2] *Sorbonne Université, CNRS, LIP6, F-75005 Paris, France*

Nous considérons un réseau utilisé pour propager des informations. Les sources d'informations fiables souhaitent que leurs messages parviennent à tous les récipiendaires sans altération. Cependant, des participants malveillants tentent de miner la crédibilité des sources en envoyant de faux messages qui semblent provenir des mêmes sources : des *fake news*.

Les solutions existantes à ce problème difficile dans un contexte réparti sont basées sur la redondance des chemins d'information nœuds-disjoints, et nécessitent pour être mises en œuvre un nombre factoriel (en la taille du réseau) de messages disséminés, et de calcul à chaque réception d'un message. Nous proposons des optimisations qui réduisent en pratique cette complexité, et nous montrons par des expérimentations sur différents types de réseaux que plusieurs ordres de grandeur peuvent être ainsi gagnés.

Mots-clefs : fake news, communication fiable, pannes Byzantines, réseau multi-sauts

1 Introduction

Nous considérons un réseau utilisé pour propager des informations. Les nœuds de ce réseau souhaitent que leurs messages parviennent à tous les récipiendaires sans altération. Cependant, des participants malveillants tentent de miner la crédibilité des sources en envoyant de faux messages qui semblent provenir des mêmes sources : des *fake news*. Ce problème est connu dans la littérature des systèmes répartis comme le problème de la *communication fiable* (ou même diffusion sécurisée ou diffusion fiable avec une source correcte) et est essentiel pour résoudre des problèmes plus complexes (tels que la diffusion fiable *sans source correcte* ou l'accord réparti). Une solution à la communication fiable garantit l'authenticité, l'intégrité et la livraison des messages échangés au sein du réseau. Ce problème ne semble pas toujours être résolu efficacement, en particulier lorsque la topologie du réseau est inconnue. L'une des solutions les plus générales pour implémenter une communication fiable a été fournie par Dolev [Dol81], et garantit l'authenticité et la livraison des messages échangés sur un réseau inconnu dans lequel un nombre limité de participants peuvent se comporter arbitrairement. Cependant, le protocole ainsi défini nécessite pour sa mise en œuvre un nombre factoriel (en la taille du réseau) de messages disséminés et un temps de calcul exponentiel à chaque réception d'un message. Aucun autre protocole, à notre connaissance, ne résout la communication fiable sans considérer des hypothèses supplémentaires (par rapport à celles qui sont nécessaires pour résoudre le problème). Nous proposons des optimisations qui réduisent en pratique cette complexité, et nous montrons par des expérimentations sur différents types de réseaux que plusieurs ordres de grandeur peuvent être ainsi gagnés. Nous présentons ici un aperçu du protocole de communication fiable et une partie des analyses, preuves et simulations que nous avons menées dans la version longue de ce travail [BFT18, BFT19].

2 Modèle

Nous supposons un ensemble fixe de n processus, chacun étant doté d'un identifiant unique. Chaque processus peut échanger des messages avec un sous-ensemble d'autres processus, ses voisins, via les canaux

[†] Giovanni Farina remercie *Université Franco-Italienne/Universitá Italo-Francese (UFI/UIF)* pour les aides à la mobilité à travers la bourse VINCI 2018.

de communication. Nous modélisons ces interactions possibles à travers un graphe $G(V,E)$ dans lequel les nœuds sont les processus et les liens correspondent aux canaux de communication disponibles. La topologie du graphe n'est pas connue des processus. Nous supposons que les messages ne sont pas perdus ou modifiés lors des échanges et que chaque processus ne peut pas mentir sur son identité lorsqu'il s'adresse à l'un de ses voisins (c'est-à-dire que nous supposons que les canaux de communication sont fiables et authentifiés). Nous supposons que dans le système, il ne peut y avoir qu'un nombre limité de processus, au plus f, qui peuvent avoir un comportement arbitraire (*pannes Byzantines*). Les autres processus sont corrects, c'est à dire qu'ils exécutent fiablement et honnêtement le code du protocole.

3 Problème de communication fiable

On apelle *source* un processus auteur d'un message qu'il souhaite diffuser à tous les processus d'un réseau. Une solution au problème de la communication fiable garantit l'authenticité, l'intégrité et la livraison des messages échangés entre les processus. Plus précisément, il doit satisfaire les spécifications suivantes.

Communication fiable - spécification du problème :

— *(sûreté)* : si un processus correct délivre un message m, alors m a été envoyé par sa source ;

— *(vivacité)* : si un processus correct envoie un message m, alors m est ultimement délivré par chaque processus correct.

Notons que satisfaire une seule de ses propriétés est trivial : il suffit de ne délivrer aucun message pour satisfaire la sûreté, et de délivrer tous les messages pour satisfaire la vivacité. C'est la satisfaction simultanée de ces deux propriétés qui pose problème.

Dolev a identifié [Dol81] la condition nécessaire et suffisante pour fournir une communication fiable dans un système réparti affecté par au maximum f pannes arbitraires. Plus précisément, la connectivité du réseau k doit être le double du nombre des processus fautifs f (c'est-à-dire $k > 2f$). Dolev a proposé la solution suivante au problème. Les messages échangés ont le format $\mathtt{m} := \langle \mathtt{s}, \mathtt{contenu}, \mathtt{visites} \rangle$ dans ce protocole : `s` est l'identifiant de la source, `contenu` est l'information contenue dans le message et `visites` est une structure de données qui conserve les identifiants des processus par lesquels le message est passé.

Protocole de Dolev :

— la source s envoie le message $\mathtt{m} = \langle \mathtt{s}, \mathtt{contenu}, \emptyset \rangle$ à tous ses voisins ;

— chaque processus p enregistre et transmet chaque message $\mathtt{m} = \langle \mathtt{s}, \mathtt{contenu}, \mathtt{visites} \rangle$ envoyé par un voisin q à tout autre voisin non inclus dans `visites` en ajoutant à `visites` l'identifiant de l'expéditeur q. Autrement dit, p enregistre et transmet $\mathtt{m} = \langle \mathtt{s}, \mathtt{contenu}, \mathtt{visites} \cup \{\mathtt{q}\} \rangle$; les messages contenant `visites` qui contient déjà l'identifiant de p sont supprimés.

— si un processus p reçoit un message $\mathtt{m} = \langle \mathtt{s}, \mathtt{contenu}, \mathtt{visites} \rangle$ de s, alors $\langle \mathtt{s}, \mathtt{contenu} \rangle$ est délivré ;

— si un processus p reçoit un ensemble de messages $\mathtt{M}_{\mathtt{p,s,contenu}} := \bigcup \mathtt{m}_\mathtt{i} = \langle \mathtt{s}, \mathtt{contenu}, \mathtt{visites}_\mathtt{i} \rangle$ portant les mêmes valeurs pour `s` et `contenu` et il existe $f+1$ ensembles nœuds-disjoints $\mathtt{visites}_\mathtt{i}$, alors $\langle \mathtt{s}, \mathtt{contenu} \rangle$ est délivré.

On peut observer que ce protocole génère un nombre factoriel de messages (par rapport à n) au sein du réseau. De plus, une instance du problème NP-Complet *set-packing* doit être résolue par chaque processus pour chaque message reçu.

4 Notre protocole

Nous présentons des modifications à introduire dans le protocole de Dolev visant à réduire le nombre de messages générés sur le réseau.

Modification 1 Il n'est pas nécessaire de garder les éléments en `visites` triés.

Modification 2 Un message $\langle \mathtt{s}, \mathtt{contenu} \rangle$ peut être délivré s'il n'y a pas f identifiants ou moins communs à toutes les `visites` liés.

Modification 3 Si un processus p a délivré $\langle \mathtt{s}, \mathtt{contenu} \rangle$, il peut jeter tous les messages $\mathtt{M}_{\mathtt{p,s,contenu}}$ associés et envoyer $\mathtt{m} = \langle \mathtt{s}, \mathtt{contenu}, \emptyset \rangle$.

Modification 4 Il faut envoyer des messages contenant $\langle \texttt{s}, \texttt{contenu} \rangle$ uniquement aux processus qui ne l'ont pas délivré.

Modification 5 Si un processus a délivré $\langle \texttt{s}, \texttt{contenu} \rangle$, il n'a qu'à envoyer $\texttt{m} = \langle \texttt{s}, \texttt{contenu}, \emptyset \rangle$.

Dans la suite, nous appelons *protocole modifié* le protocole de Dolev auquel les modifications 1-5 sont apportées.

Theorem 1. *[BFT19] Le protocole modifié garantit une communication fiable dans tous les systèmes où une communication fiable est assurée par le protocole de Dolev.*

Chaque processus peut collecter plusieurs messages dans $M_{p,s,contenu}$, qui à leur tour seront transmis aux processus voisins. Cependant, tous les messages de $M_{p,s,contenu}$ ne sont pas nécessaires pour pouvoir délivrer $\langle \texttt{s}, \texttt{contenu} \rangle$. Pour cette raison, les processus prioriseront donc les messages $\texttt{m} = \langle \texttt{s}, \texttt{contenu}, \texttt{visites} \rangle$ contenant des `visites` plus courts.

5 Évaluations expérimentales

Nous testons l'efficacité du protocole modifié en considérant le nombre de messages échangés dans le système. Nous simulons des instances individuelles des algorithmes de communication fiable dans un système synchrone où soit f processus aléatoires ne transmettent pas de messages (*processus fautifs silencieux*), soit ils les envoient avec des champs `visites` ne correspondants pas à la réalité (*processus fautifs actifs*). Nous considérons trois types de graphes différents : Barabási-Albert, aléatoire k-régulier k-connexe et roue multipartite. Les graphiques présentés en figure 1 montrent que : *(i)* le nombre de messages échanges est considérablement réduit avec notre algorithme comparé au protocole de Dolev (a) ; *ii)* la quantité de messages effectivement générée dépend de la topologie du réseau (b,c,d) et il existe des topologies, comme la roue multipartite, où les modifications proposées ont une efficacité variable (d) ; *iii)* les performances ne se détériorent pas si les processus fautifs relaient les messages avec des champs `visites` incorrects (e,f). De plus amples détails sur les simulations individuelles sont fournis dans la légende de la figure 1.

6 Conclusion

Nous sommes partis d'un réseau utilisé par ses participants pour diffuser des informations dont le but était d'empêcher la propagation d'informations malveillantes générées par un nombre limité de nœuds. Nous avons présenté l'une des solutions les plus générales à ce problème, le protocole de Dolev, qui ne peut cependant pas être utilisée efficacement au sein du réseau en raison du nombre de messages générés très important. Nous avons proposé des modifications à cette solution et nous avons montré à travers des simulations que ces modifications sont efficaces pour réduire le nombre de messages échangés sur le réseau sur différentes topologies, typiquement, on passe d'un nombre factoriel de messages échangés à un nombre seulement quadratique. Il reste un problème ouvert : savoir s'il est toujours possible de fournir une communication fiable de manière efficace quelle que soit la topologie du réseau, c'est-à-dire avec un nombre polynomial de messages de la taille du réseau, sans considérer d'autres hypothèses par rapport à celles identifiées par Dolev. La réponse pourrait ouvrir la possibilité d'une communication fiable et efficace même sur des réseaux qui évoluent dans le temps.

Références

[BFT18] Silvia Bonomi, Giovanni Farina, and Sébastien Tixeuil. Multi-hop byzantine reliable broadcast made practical. In *8th Latin-American Symposium on Dependable Computing, LADC 2018, Foz do Iguaçu, Brazil, October 8-10, 2018*, pages 155–160. IEEE, 2018.

[BFT19] Silvia Bonomi, Giovanni Farina, and Sébastien Tixeuil. Multi-hop byzantine reliable broadcast with honest dealer made practical. *J. Braz. Comp. Soc.*, 25(1) :9 :1–9 :23, 2019.

[Dol81] Danny Dolev. Unanimity in an unknown and unreliable environment. In *22nd Annual Symposium on Foundations of Computer Science, Nashville, Tennessee, USA, 28-30 October 1981*, pages 159–168. IEEE Computer Society, 1981.

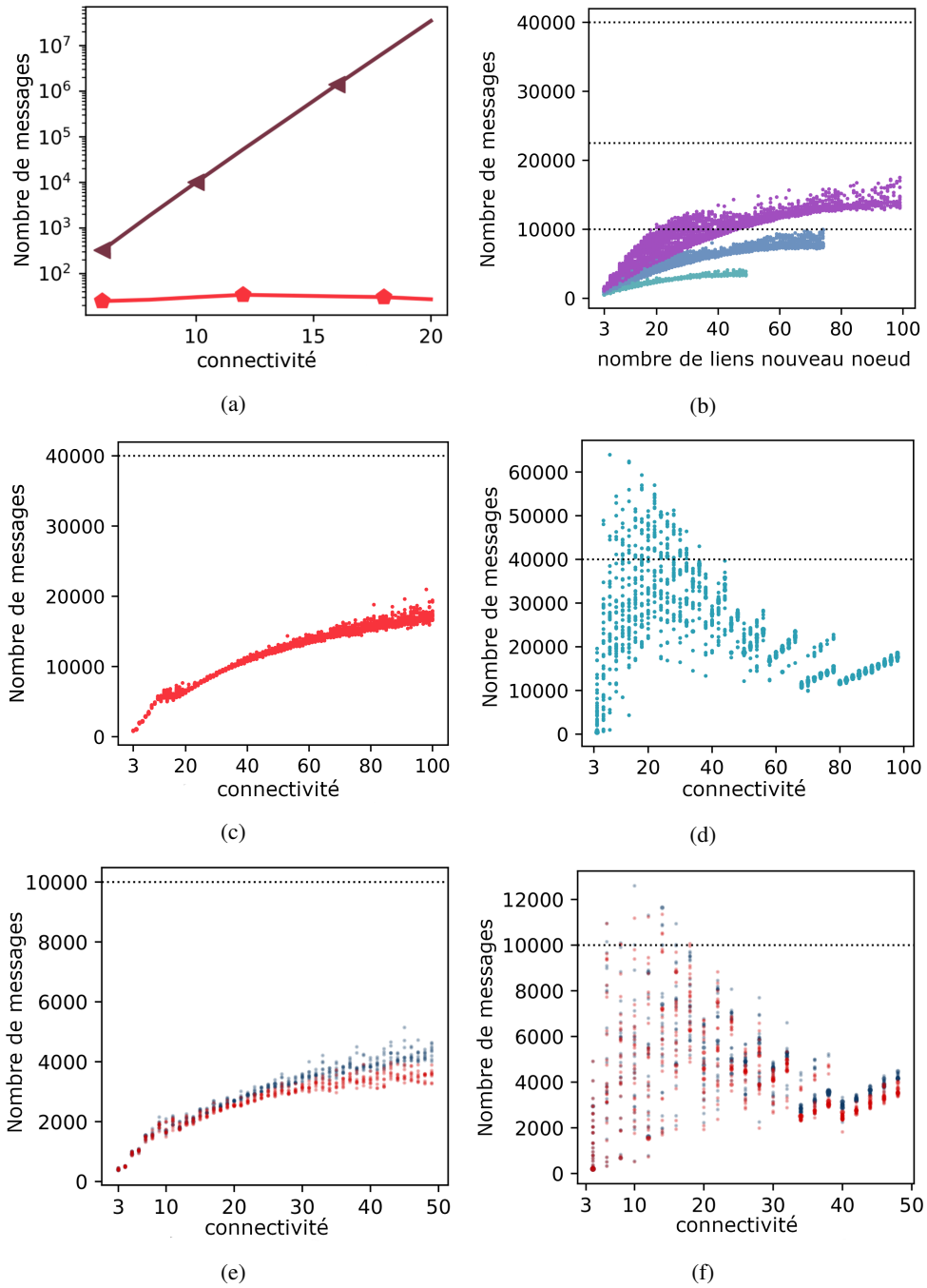

FIGURE 1: (a) graphe aléatoire k-régulier, comparaison entre Dolev (en marron) et le protocole modifié (en rouge), processus fautifs silencieux ; (b) graphe Barabási-Albert graphe, $n = 100, 150, 200$ (respectivement de bas en haut), protocole modifié, processus fautifs silencieux ; (c) graphe aléatoire k-régulier k-connexe, $n = 200$, protocole modifié, processus fautifs silencieux ; (d) roue multipartite, $n = 200$, protocole modifié, processus fautifs silencieux ; (e) graphe aléatoire k-régulier k-connexe, $n = 100$, protocole modifié, processus fautifs silencieux (en bleu) et actifs (en rouge) ; (f) roue multipartite, $n = 100$, protocole modifié, processus fautifs silencieux (en bleu) et actifs (en rouge). Pour les figures (b)-(f), la ligne horizontale pointillée réfère à un nombre quadratique de messages échangés.